

AMENDMENTS TO THE CLAIMS

All pending claims are reproduced below:

1. (Currently amended) A method of creating an application for executing on at least one machine having a memory, the method comprising:
 - creating a definition of at least one node and a specification, which are held in at least one machine readable data file;
 - the specification defining how the at least one node ~~can~~ interacts with other nodes during processing of the machine readable data file, resources useable by the at least one node during processing of the machine readable data file, at least one set of predetermined rules to be used by the node for processing data and messages which are arranged to ~~that can~~ be passed between nodes such that the node is arranged to receive ~~capable of receiving~~ messages ~~in any format~~ which ~~can~~ trigger a rule if predetermined data is present;
 - providing a run time environment which processes the ~~at least one~~ machine readable data file in order to implement the at least one node within the memory of the machine such that the node arranged to receive ~~is capable of receiving~~ data, ~~and processing~~ process the received data according to the set of predetermined rules and output ~~outputting~~ the processed data;
 - wherein the processing of the machine readable data file, in the run time environment, dynamically interconnects each node according to the ~~definition provided in the specification~~ and/or a data input such that data input to the application is processed by the at least one node and, if further processing is required, forwarded to other nodes for that further processing.
2. (Currently amended) The [[A]] method according to claim 1 in which a plurality of nodes are created.
3. (Currently amended) The [[A]] method according to claim 1 which further comprises providing a library of nodes containing at least one node and selecting at least one of the nodes from the library of nodes.

4. (Currently amended) The [[A]] method according to claim 1 which further comprises arranging the or each node to comprise a plurality of layers, each layer being arranged to perform a predetermined function.

5. (Currently amended) The [[A]] method according to claim 4 which further comprises arranging the layers of the nodes to be interchangeable and wherein altering at least one of the layers can change the overall functionality of a node.

6. (Currently amended) The [[A]] method according to claim 4 which further comprises providing a library of layers containing at least one layer and selecting at least one layer from the library of layers.

7. (Currently amended) The [[A]] method according to claim 4 which comprises arranging at least one of the layers of a node to act as a transport layer arranged to receive and send data to and from the node.

8. (Currently amended) The [[A]] method according to claim 4 which comprises arranging at least one of the layers of a node to act as a message transceiver arranged to send and receive messages to other nodes to which that node is connected.

9. (Currently amended) The [[A]] method according to claim 8 in which the at least one node has an identity and in which the application is arranged to be run and, at runtime, as nodes are connected together, the method further comprising arranging the message transceiver layer of a node to discover the identity of nodes to which it is connected at runtime.

10. (Currently amended) The [[A]] method according to claim 4 which comprises arranging at least one of the layers of a node to act as a rule processing engine arranged to apply the predetermined rules to data that the node receives.

11. (Currently amended) The [[A]] method according to claim 10 in which the rule processing engine layer of a node uses forward chaining rule logic.

12. (Currently amended) The [[A]] method according to claim 10 which comprises providing a rule set of at least one rule in a file that is used by the rule processing engine.

13. (Currently amended) The [[A]] method according to claim 12, which comprises specifying the file in which the rules are located by a link.

14. (Currently amended) The [[A]] method according to claim 10 which comprises defining each rule set that is to be used by the application in the specification.

15. Canceled.

16. Canceled.

17. (Currently amended) The [[A]] method according to claim 1 which comprises writing the messages in a flat text format, which may be any of the following: ASCII, XML, EDI (Electronic Data Interchange).

18. Canceled.

19. Canceled.

20. (Currently amended) The [[A]] method according to claim 1 which comprises providing a pattern, arranging the at least one node within the pattern and defining how nodes therein interact with one another.

21. (Currently amended) The [[A]] method according to claim 20, which comprises providing a library of patterns containing at least one pattern that can be used in creating an application.

22. (Currently amended) The [[A]] method according to claim 1 in which the specification is arranged to determine at least one of the following: which nodes are to be used; which nodes interact with one another; which patterns are to be used; which assets are to be used.

23. (Currently amended) The [[A]] method according to claim 1 which comprises providing files arranged to define the application specified therein, arranging the specification to be capable of deploying files and using the specification to deploy the files.

24. (Currently amended) The [[A]] method according to claim 23 which comprises arranging the files specifying the application to be XML files.

25. (Currently amended) The [[A]] method according to claim 1 in which the data processed by the application is specified in an XML file.

26. (Currently amended) The [[A]] method according to claim 1 in which data processed by the application is specified in an image file.

27. (Currently amended) The [[A]] method according to claim 1 in which data processed by the application is specified in a flat text file such as an ASCII file, a raw text file, and EDI file.

28. (Currently amended) The [[A]] method according to claim 1 which comprises providing a graphical tool arranged to enable a user to specify components of the application.

29. (Currently amended) The [[A]] method according to claim 28 which comprises providing a library of at least one of the following: nodes; node layers; specification; patterns; messages; rule sets; style sheets; schemas and in which the graphical tool allows a user to select components from one of said libraries.

30. (Currently amended) The [[A]] method according to claim 29 which allows a user to define further libraries.

31. (Currently amended) The [[A]] method according to claim 28 which comprises providing at least one pattern arranged to define how nodes interact arranging the at least one pattern such that it is capable of interacting with at least one other pattern and arranging the graphical tool to allow a user to specify how the patterns and nodes interact with one another.

32. (Currently amended) The [[A]] method according to claim 28 which comprises using the graphical tool to perform at least one of the following: create the specification; edit the specification.

33. (Currently amended) The [[A]] method according to claim 28 which comprises using the graphical tool to manipulate any components of the specification.

34. (Currently amended) The [[A]] method according to claim 1 which comprises creating and deploying files and processing the files in the run time environment.

35. (Currently amended) The [[A]] method according to claim 1 in which at least one node is provided to provide an output from the application.

36. (Currently amended) A computer system having a memory and being arranged to create an application, said system comprising:

- a node creator arranged to create a definition ~~specifying of~~ at least one node and a specification, and for each node specified, the ~~definition~~ specification defining how that node will interact with any other nodes during processing of a machine readable data file, resources useable by that node during processing of the machine-readable data file, a set of predetermined rules to be used by that node for processing data and messages that ~~can~~ are arranged to be passed between that node and any other nodes such that that node is ~~capable of receiving~~ arranged to receive messages ~~in any format~~ which can trigger a rule if predetermined data is present, the node being ~~capable of processing~~ arranged to process data according to [[a]] the set of predetermined rules and ~~generating~~ generate an output therefrom;
- a linker ~~capable of connecting~~ arranged to connect, dynamically according to the specification and/or any data input to the application, at least two nodes such that data ~~can~~ is arranged to pass between the nodes and the linker being arranged to interact with the node creator to modify the definition;
- a deployer arranged to deploy the application from the definition created by the node creator and the linker according to a specification.

37. (Currently amended) The [[A]] computer system according to claim 36 which comprises at least one processor arranged to process data, including the definition, and on which the definition created by the node creator and modified by the linker is processed.

38. (Currently amended) The [[A]] computer system according to claim 37 which comprises at least one processing apparatus comprising the at least one processor and in which the linker is arranged to connect nodes running on the processor within the processing apparatus.

39. (Currently amended) The [[A]] computer system according to claim 38 which comprises a plurality of processors, each remote from the other and having a connector therebetween capable of transmitting data between the processors.

40. (Currently amended) The [[A]] computer system according to claim 39 in which each of the processors is provided on a separate processing apparatus.

41. (Currently amended) The [[A]] computer system according to claim 39 in which the linker is arranged to connect nodes provided on processors remote from one another.

42. (Currently amended) The [[A]] computer system according to claim 36 in which the deployer deploys the definition that causes the nodes to communicate with one another using one of HTTP and direct memory protocols.

43. (Currently amended) The [[A]] computer system according to claim 36 in which the node creator is arranged to utilise at least one of the following: predetermined definitions and pre-written definitions.

44. (Currently amended) The [[A]] computer system according to claim 43 in which the pre-written definition is provided in at least one library.

45. (Currently amended) The [[A]] computer system according to claim 36 which further comprises a pattern creator arranged to create at least one pattern of nodes.

46. (Currently amended) The [[A]] computer system according to claim 36 which further comprises a pattern cloner arranged to clone a pattern of nodes.

47. (Currently amended) The [[A]] computer system according to claim 36 which further comprises a rule creator arranged to allow predetermined rules to be created and edited.

48. (Currently amended) The [[A]] computer system according to claim 36 which further comprises at least one of the following: a node storage; a pattern storage; a rule storage.

49. (Original) A machine readable medium containing instructions which when read onto a computer cause that computer to perform the method of claim 1.

50. (Original) A machine readable medium containing instructions which when read onto a computer cause that computer to function as the computer system according to claim 36.

51. Canceled.

52. Canceled.

53. Canceled.

54. Canceled.

55. Canceled.

56. Canceled.

57. Canceled.

58. Canceled.

59. Canceled.

60. Canceled.

61. Canceled.

62. (Currently amended) The [[A]] method according to claim 1 in which the node, specification, and messages are at least in part written in XML.

63. (Currently amended) A computer system having a memory and being ~~arrange~~ arranged to run an application, said system comprising

a run time environment arranged to process at least one machine readable data file, the machine readable data file providing a definition of at least one node and a specification, the specification defining how the at least one node ~~can~~ is arranged to interact with other nodes during processing of the machine readable data file, resources useable by the at least one node during processing of the machine readable data file, at least one set of predetermined rules to be used by the node for processing data and messages that ~~can~~ are arranged to be passed between nodes such that a node is ~~capable of receiving~~ arranged to receive messages ~~in any format~~ which ~~can~~ trigger a rule if predetermined data is present;

wherein the run time environment comprises a linker which is arranged to connect, dynamically according to the specification and/or any data input to the application, the at least one node to any other nodes such that data input to the application is processed by the at least one node and, if further processing is required, forwarded to the other nodes for that further processing.

64. (Previously presented) A machine readable medium containing instructions which when read onto a computer cause that computer to function as the computer system according to claim 63.

65. (New) The method of claim 1, wherein the or each node is arranged to manipulate data contained in a message.

66. (New) The method of claim 65, wherein the or each node is arranged to manipulate XML data contained in the message.

67. (New) The method of claim 1, wherein the or each node is arranged, during processing of the machine readable data file, to output data to any of the nodes to which it is arranged to be connected.